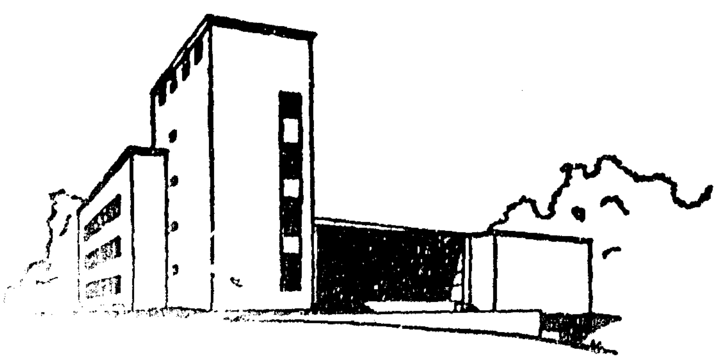
Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Graduate School of Industrial Administration Carnegie-Mellon University	2a. REPORT SECURITY CLASSIFICATION Unclassified
	2b. GROUP Not applicable

3. REPORT TITLE

THE FACIAL DECOMPOSITION METHOD

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Management Sciences Research Report May 1972

5. AUTHOR(S) (First name, middle initial, last name)

Claude-Alain Burdet

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1972	10	8

8a. CONTRACT OR GRANT NO. N00014-67-A-0314-0007	9a. ORIGINATOR'S REPORT NUMBER(S) Management Sciences Research Report No. 280
b. PROJECT NO. NR 047-048	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) W.P. 104-71-2
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Logistics and Mathematical Statistics Br. Office of Naval Research Washington, D. C. 20360

13. ABSTRACT

This note presents a brute force approach to linearly constrained programming in non-convex optimization; our aim here is to illustrate a general methodology which can be applied to construct tailor-made algorithms in specific applications.

In essence, the facial decomposition method constructs a non-redundant list of all faces of the polyhedral set $P \subset R^n$. Each face is characterized by a linear program in a given affine subspace of R^n. This list is conveniently displayed in a tree structure which represents the set of nodes to be searched (typically for optimality).

Unclassified
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Branch and Bound						
Implicit Search						
Linear Programming						
Non-Convex Programming						

DD FORM 1 NOV 65 1473 (BACK)

Unclassified

W.P. 104-71-2

Management Sciences Research Report No. 280

The Facial Decomposition Method

by

Claude-Alain Burdet

May 1972

This report was prepared as part of the activities of the Management Sciences Research Group, Carnegie-Mellon University, under Contract N0014-67-A-0314-0007 NR 047-048 with the U. S. Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

This note presents a brute force approach to linearly constrained programming in non-convex optimization; our aim here is to illustrate a general methodology which can be applied to construct tailor-made algorithms in specific applications.

In essence, the facial decomposition method constructs a non-redundant list of all faces of the polyhedral set $P \subset \mathbb{R}^n$. Each face is characterized by a linear program in a given affine subspace of R^n. This list is conveniently displayed in a tree structure which represents the set of nodes to be searched (typically for optimality).

The Facial Decomposition Method

by Claude-Alain Burdet

0) Generalities

This note presents a brute force approach to linearly constrained programming which has proved successful in non-convex optimization for problems of moderate size; our aim here is to illustrate a general methodology which (like the branch and bound idea, for instance) can be applied to construct tailor-made algorithms in specific applications.

In essence, the facial decomposition method constructs a non-redundant list of all the k-dimensional $(0 \leq k \leq n)$ faces of the polyhedral set $P \subset \mathbb{R}^n$, [1]. Each face is characterized by a linear program in a given affine subspace of $\mathbb{R}^n$.

The set of all faces of a polyhedron P is a complete lattice and, in particular, one has the property:

> A face of a face F of P is a (sub)face of P; thus, the set inclusion induces a finite tree structure on the set of all feasible solutions (i.e., P), each node corresponding to a face of P.

A method is presented in [1] to generate a non-redundant list of linear programs, each belonging to a face of P. This list is conveniently displayed in a tree structure which represents the set of nodes to be searched (typically for optimality); this constitutes the face decomposition method.

Unlike branch and bound which is a two criterion (feasibility and optimality) search, the facial decomposition method represents a one

<u>criterion tree search approach</u> where feasibility with respect to P is automatically satisfied.

In practice the face structure of a polyhedron is an elusive concept because it may be radically altered by the influence of numerical errors; typically numerical perturbations due to the finite accuracy of the computations will introduce a multitude of additional "faces" whenever degeneracy occurs. A similar phenomenon may also be present due to inaccuracy of the original data. In order to remedy this situation, the facial decomposition approach makes use of three concepts: <u>relevant</u>, <u>pseudo-degenerate</u>, and <u>redundant inequalities</u>.

In fact this trichotomy introduces a <u>control parameter</u> which can be used to limit the size of the tree; one may in this manner guide the search within a subtree towards rapidly finding a <u>good solution</u>. In a second effort, global optimality can be attained by searching the rest of the tree.

Applications of the facial decomposition method are found in (optimization) problems of various types; in order to be decomposed a problem must possess the following property π :

> The relative interior rel int F of an arbitrary face F of P can always be fathomed (i.e., discarded from further consideration).

For instance in concave, and in zero-one programming, it is known <u>a priori</u> that the optimal solution is an extreme point of P and therefore not interior to any face. Another application for quadratic programming is presented in [2].

1) <u>Minimal sets</u>

The idea behind the construction of facial decompositions is quite simple; it rests on the fact that a linear program with variables $x_1,\dots,x_n \geq 0$ (including slacks) may contain <u>redundant</u>, <u>degenerate</u>, and <u>relevant</u> variables.

Let $Ax = b$, $x \geq 0$ (1)

characterize the linear program;

<u>Definition 1</u>: The variable x_i is called <u>redundant</u> if $\Delta > \sigma_i^+ \geq 0$ with $\Delta = \min x_i$ subject to $Ax = b$, $x_j \geq 0, \forall_j \in J$ where the set J is chosen to satisfy $J \subset N$ .... $i \notin J$. (2)

<u>Definition 2</u>: The variable x_i is called <u>pseudo-degenerate</u> if $-\sigma_i^- \leq \Delta \leq \sigma_i^+$, with $\sigma_i^- > 0$

<u>Definition 3</u>: The variable x_i is called <u>relevant</u> otherwise, i.e. if $\Delta < \sigma_i^- \leq 0$.

<u>Remarks</u>:

1) Definitions 1, 2 and 3 are critically dependent on the set $J \subset N$, and one should speak of <u>relevant variables with respect to</u> J , for instance.
2) If we assume the arithmetic to be exact, one may set $\sigma_i^+ = \sigma_i^- = 0$.
3) The partition of the set N into three complementary subsets $I_{redundant}$, $I_{pseudo\text{-}degenerate}$, $I_{relevant}$, is not unique; this is not a burden, however, because the facial decomposition merely requires the existence (and choice) of one such partition.

Definition 4: A set $I \subset N$ is called minimal if one has

$$\{x \mid Ax = b,\ x_i \geq 0\ ,\ \forall i \in I\} \subset \{x \mid Ax = b,\ x_i \geq 0\ ,\ \forall i \in N\}$$

and each variable x_i, $i \in I$ is relevant with respect to the set $(I - \{i\})$.

By definition a minimal set of non-negative variables characterizes the same feasible region as the original L.P.; furthermore, if one assumes that this region is a p-dimensional (polyhedral) set P , then each minimal variable x_i, $i \in I$ identifies a (p-1)-dimensional facet F of P which can be characterized by a linear program obtained from the previous one by setting $x_i = 0$; upon determination of a minimal set for this facet F (it is a subset of I), one will identify (p-2)-dimensional subfaces of P ; and so on...; this process is easily seen to generate the finite set of all faces of P in a tree-like manner. More details concerning this construction can be found in [1].

For any face, the definition of the minimal set can be based on control parameters σ_i^+, σ_i^- which may differ from zero; this will produce smaller minimal sets, whose size clearly depends on the numerical value of σ_i^+ and σ_i^- . For small enough values of σ_i^+ and σ_i^- , these parameters will merely compensate for the numerical inaccuracy of the computations and only those "faces" which have been "created" by numerical perturbations will be discarded; for larger values of σ_i^+ and/or σ_i^- , however, only a subset of the actual faces of P will be generated. In fact such control parameters introduce a choice criterion among the faces of P , since only certain faces will be generated in the tree; the selected few, however, can be seen to range through all dimensions and across the entire tree.

Of course, in this case,(since not all the faces are tested, say, for optimality) one will, in general, merely obtain a good (usually suboptimal) solution. As in all search procedures there are a number of strategies at our disposition: Typically a good solution should be rapidly delivered by the algorithm (a variety of heuristic guidelines for the choice of σ_i^+ and σ_i^- for each $i \epsilon N$ can be used); in a second phase, it is often desirable to spend some more computing time, trying to establish global optimality, or to improve the current best solution; this is easily accomplished in the facial decomposition method by generating some new faces which correspond to variables which have previously been classified as pseudo-degenerate, during the preceding approximations. (This amounts in effect to lowering the value σ^-.) The minimal sets need not be recomputed, but merely need be extended by adjoining to them some pseudo-degenerate variables; since all index sets are finite, the search will eventually terminate.

It would be futile to list all possible algorithms and strategies based on facial decomposition; the method is simple and flexible enough to lend itself to special structures as well as to unstructured problems.

In zero-one programming, the facial decomposition approach can be seen to be intimately related to the classical branch and bounds methods because of the elementary constraints $0 \leq x_i \leq 1$ (the actual comparison is left to the reader as an exercise).

Thus the facial decomposition may be viewed as an extension towards less structured problems. Possible domains of application in concave, and integer programming are sketched in [1]. A detailed algorithm for the general quadratic algorithm is presented in [2] and extended in [3]. Experimental codes seem to indicate that the approach is powerful enough to motivate further analysis both theoretical and applied.

2) <u>Cutting planes</u>

The tree search and the size of the facial tree can be considerably curtailed, if one introduces (at an arbitrary node) some additional extraneous linear constraint(s); <u>valid</u> or <u>enumerative cutting planes</u> can be used for this purpose. Enumerative cuts are <u>conditionally valid</u>, in the sense that they require some additional implicit search to determine the best feasible solution which they may cut off; any of the well-known search methods (typically branch and bound) can be used for this purpose; thus one may define combined algorithms which mesh facial decomposition and branch and bound into one another. Here facial decomposition assumes the role of a monitoring device which remedies the inherent difficulty of branch and bound to cope with the problem of feasibility.

At first, it may seem that the introduction of new cuts renders the structure of the feasible polyhedral set more complex by creating a host of new faces. This is indeed the case, but the situation is automatically taken care of by allowing only <u>original variables</u> (i.e., not the <u>cut variables</u>) to enter minimal sets.

<u>Definition 4'</u>: A set $I \subset N$ is called minimal if one has:

$$\{x' \mid A'x' = b',\ x'_i \geq 0\ ,\ \forall i \in I \cup C\} \subset \{x' \mid A'x' = b',\ x'_i \geq 0\ ,\ \forall i \in N \cup C\}$$

and each variable x_i, $i \in I$ is <u>relevant</u> with respect to the set $(I - \{i\}) \cup C$,

where $A'x' = b'$ represent an extension of the original L.P, where cuts have been implemented; C is the index set of the cut variables.

Hence the net effect of cutting planes is to reduce the feasible set (and the facial tree) without introducing new nodes. The justification of this approach is that a "cut face" lies entirely in the face F of P corresponding to the node for which the cut was implemented; therefore each point of the cut face lies either in the relative interior of F (and according to the property π it can be discarded) or it belongs to the relative boundary of F and it is considered explicitly in the further ramifications of the facial tree.

3) Conclusions

We indicate that both the cutting plane and implicit search methods can be profitably imbedded into the facial decomposition approach; this results in a reduction of the size of the facial tree, without any additional bookkeeping complications. Preliminary computational experimentation seems to indicate that the monitoring scheme offered by the facial decomposition also reduces the size of implicit search and increases the depth and efficiency of cutting planes. The computational efficiency of a method of this type is not easily tractable, because software considerations usually play an important role; furthermore the success of methods in the area of non-convex mathematical programming is usually highly "problem dependent." However the following heuristic arguments can be put forward to validate the facial decomposition approach:

- because implicit search is performed within a simplex (in order to prove the validity of an enumerative cut) it usually is quite efficient.
- because a cutting plane can be generated in a lower dimensional face, its depth (within that face) is always at least as good and usually better than if the cut were generated in a higher dimensional affine space; this is particularly true when degeneracy occurs.
- because a non-relevant variable generates a whole subtree in the full factorial design of all faces (feasible or not) of P , one may expect an exponential payoff in return for the additional pivoting required by the construction of minimal sets.

- when the algorithm terminates, it does not only deliver an optimal (or currently best) solution but also the necessary elements for sensitivity analysis in the form of a description of the facial structure about this solution.

The above hortative documentation could certainly be extended in many other directions, especially if the intrinsic characteristics of various applications are taken into account. The object of this note, however, is not to claim superior numerical results but primarily to indicate a line of research; the facial decomposition approach seems promising enough to motivate a quest for more results, particularly for large and/or structured systems.

References

[1] Burdet, C.-A.: "Generating All the Faces of a Polyhedron," Management Science Report #271, Graduate School of Industrial Administration, Carnegie-Mellon University, submitted to the SIAM Journal on Applied Mathematics (1972).

[2] _____________: "General Quadratic Programming," Management Science Report #272, Graduate School of Industrial Administration, Carnegie-Mellon University, submitted to Management Science (1972).

[3] _____________: "Some Enumerative Inequalities and Their Application in Integer Programming Algorithms," published in Mathematical Programming, Vol. 2, No. 1, (1972) pp. 32-64.

[4] _____________: "Enumerative Cuts I and II," submitted to Operations Research, (1970 and 1972 respectively).

[5] _____________: "Polaroids: a new tool in non-convex and in integer programming," W.P. 78-72-1, Graduate School of Industrial Administration, Carnegie-Mellon University, submitted to Naval Research Logistics Quarterly, (1972).

[6] _____________: "Simple Polaroids for Non-convex Quadratic Programming," W.P. 101-71-2, Graduate School of Industrial Administration, Carnegie-Mellon University, (1972).

[7] Tuy, Hoang: "Concave Programming Under Linear Constraints," (Russian), Doklady Akademii Nauk SSSR, 1964. English translation in Soviet Mathematics, 1964, pp. 1437-1440.

[8] Glover, R: "Convexity Cuts", University of Texas, Austin, Dec. 1969.